

# Package ‘wgaim’

September 3, 2014

**Type** Package

**Title** Whole Genome Average Interval Mapping for QTL detection using mixed models

**Version** 1.4-3

**Date** 2014-09-03

**Author** Julian Taylor <julian.taylor@adelaide.edu.au>, Simon Diffey <simon.diffey@dpi.nsw.gov.au>, Ari Verbyla <ari.verbyla@csiro.au> and Brian Cullis <bcullis@uow.edu.au>.

**Maintainer** Julian Taylor <julian.taylor@adelaide.edu.au>

**Depends** R (>= 2.0.0), qtl, lattice

**SystemRequirements** asreml-R 3.x

**Description** This package integrates sophisticated mixed modelling methods with a whole genome approach to detecting significant QTL in linkage maps.

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-09-03 08:45:24

## R topics documented:

wgaim-package . . . . .	2
cross2int . . . . .	4
genoCxR . . . . .	6
genoRxK . . . . .	7
genoSxT . . . . .	7
link.map.cross . . . . .	8
link.map.default . . . . .	10
link.map.wgaim . . . . .	12

out.stat . . . . .	14
phenoCxR . . . . .	16
phenoRxK . . . . .	16
phenoSxT . . . . .	17
qtlTable . . . . .	18
summary.wgaim . . . . .	20
tr.wgaim . . . . .	22
wgaim.asreml . . . . .	23

<b>Index</b>	<b>27</b>
--------------	-----------

---

wgaim-package	<i>Whole Genome Average Interval Mapping (wgaim) for QTL detection</i>
---------------	--

---

## Description

This package uses sophisticated mixed modelling methods with the addition of allowing a whole genome approach to detecting significant QTL in linkage maps.

## Details

Package: wgaim  
 Type: Package  
 Version: 1.4  
 Date: 2013-04-23  
 License: GPL 2

Welcome to version 1.x of wgaim! The documentation given in this help file is only brief and users should eventually consult the vignette available with the package by typing `vignette("wgaim")` at prompt.

Version 1.x package highlights:

1. The inclusion of high dimensional genetic data.
2. Users have a choice of whole genome marker or interval analysis.
3. Selection and estimation of QTL effects using the implementation of Verbyla et. al (2012).

Deprecated functions:

1. `read.interval`: This function used to read in and convert genetic marker data to "interval" objects ready for analysis by wgaim. Users are now directed to use `read.cross` from the **qtl** package and convert to an "interval" object by using the conversion function `cross2int`.
2. `wmerge`: This function seamlessly merged phenotypic and "interval" genetic data. This is not required as the merging is contained within the wgaim call.

This package builds on the **qtl** package of Broman by including additional functions for whole genome QTL analysis of a full linkage map using linear mixed models.

The package provides a user friendly function `cross2int` for the conversion of "cross" objects created using `read.cross` in Bromans `qtl` package into an "interval" object or use in `wgaim`. Specifically, `cross2int` performs additional calculations to impute missing marker values on each of the chromosomes across the full linkage map and also provides users with genetic distances and recombination fractions for the intervals. The returned object retains the class structure of an object created with `read.cross` and thus allows further use with the `qtl` package if desired.

The package also provides a very neat graphical display of the chromosomes of a "cross" object. The method function `link.map` displays the full or subsetted linkage map according to chromosome or distance as well as displays non-overlapping marker names on the right hand side.

QTL analysis of the is achieved using the function `wgaim` which, as its first argument, requires an `asreml` base model. Version 1.x of `wgaim` allows users to include high dimensional genetic components in a `wgaim` analysis (See `wgaim.asreml` for more details or the soon to be released vignette companion to this version). For convenience the default tracing of results from the `asreml` models is outputted to a file for further inspection. For diagnostic purposes, the outlier statistics from each iteration can be viewed using `out.stat`. Diagnostics of the likelihood ratio test performed for each forward step can be displayed using `tr.wgaim`. The function also displays an incremental probability value matrix of the QTL ascertained at each forward step of the algorithm.

Summary and print methods are available for the returned "wgaim" object and provide users with a detailed report on the QTL, their size, their flanking markers, significance (including LOD score if desired) and approximate contribution to the genetic variance. The returned "wgaim" object may also be plotted using the method function `link.map`. This function plots the full linkage map subsetted for chromosome and distance as well as provides shaded QTL regions and highlighted flanking markers. Plotting of QTL for multiple traits is also possible (see `link.map.default`)

### Author(s)

Julian Taylor, Simon Diffey, Ari Verbyla and Brian Cullis Maintainer: Julian Taylor <julian.taylor@csiro.au>

### References

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL <http://www.jstatsoft.org/v40/i07/>.

Verbyla, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical And Applied Genetics*, **116**, 95-111.

### See Also

[qtl-package](#)

cross2int

*Convert a cross genetic object to an interval object***Description**

Converts an object of class "cross" to an object with class "interval". The function also imputes missing markers.

**Usage**

```
cross2int(fullgeno, missgeno = "MartinezCurnow", rem.mark = TRUE,
          id = "id", subset = NULL)
```

**Arguments**

fullgeno	an object of class "cross" with restrictions (see Details)
missgeno	a character string determining how missing values in the linkage map should be imputed. If "Broman", then missing values are imputed according to Broman's rules. If "MartinezCurnow" then missing values are imputed according to the rules of Martinez & Curnow (1994) (see reference list). The default is "MartinezCurnow" (see Details).
rem.mark	logical value. If TRUE co-locating markers are deleted and placed in the component of the object (see Details). Defaults to TRUE.
id	a character string or name of the unique identifier for each row of genotype data (see Details). Defaults to "id"
subset	a possible character vector naming the subset of chromosomes to be returned. Defaults to NULL implying return all chromosomes.

**Details**

This function provides the conversion of genetic data objects that have already been created using `read.cross` from Broman's **qtl** package to "interval" objects ready for use with `wgaim`.

User should be aware that this function is restricted to populations with only two genotypes. `fullgeno` is checked for the class structure `c("bc", "cross")`. If this is not present an error is returned.

During the conversion process 3 important linkage map calculations are checked.

1. The map may be subsetted using the `subset` argument
2. If `rem.mark = TRUE` then markers that are co-located are removed before missing values are imputed. This marker similarity is determined by the genetic distances that are given in the map for each linkage group. If a set of markers co-locate the name of the first marker is chosen and a single consensus marker is determined by coalescing the genetic information from all markers in the set. A "(C)" is placed after the marker name for easy identification. The markers removed from each set is returned with the object and placed under `"cor.markers"` for inspection if required.

- Missing values are imputed according to the argument given by `missgeno`. This imputation results in a complete version of the marker data for each chromosome which is then used to create the interval data "intval". The complete marker data for each chromosome can be obtained from the "imputed.data" element of the returned list. It is therefore also possible to perform whole genome *marker* analysis using `wgaim`. See `wgaim.asreml` for more details.

Note: this last step is crucial as a complete set of marker or interval data is required for analysis with `wgaim`.

### Value

a list of class "cross" that also inherits the class "interval". The list contains the following components

<code>geno</code>	This is a list with elements named by the corresponding names of the chromosomes. Each chromosome is itself a list with six elements: "data" is the actual estimated map matrix with rows as individuals named by "id" and markers as columns; "map" is a vector of marker positions on the corresponding chromosome; "imputed.data" is identical to "data" matrix but with all NA's replaced by imputed values according to the rules of "missgeno"; "dist" contains the genetic distance between adjacent markers or the genetic distances of the intervals; "theta" contains the recombination fractions for each interval; "intval" contains the recalculated intervals based on the recombination fractions and the missing marker information.
<code>cor.markers</code>	If <code>rem.mark = TRUE</code> , a three column matrix with each row describing which pairwise markers are correlated and what chromosome they are from.
<code>pheno</code>	A data.frame of phenotypic information with rows as individuals read in from <code>read.cross</code> . A copy of the column named by the "id" argument can be found here (see <code>read.cross</code> ).

### Author(s)

Julian Taylor, Simon Diffey, Ari Verblya and Brian Cullis

### References

- Martinez, O., Curnow, R. N. (1994) Missing markers when estimating quantitative trait loci using regression mapping. *Heredity*, **73**, 198-206.
- Julian Taylor, Arunas Vebyla (2011). R Package `wgaim`: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL <http://www.jstatsoft.org/v40/i07/>.
- Verblya, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical And Applied Genetics*, **116**, 95-111.

### See Also

[read.cross](#)

## Examples

```
## Not run:
# read in linkage map from a rotated .CSV file with "id" as the
# identifier for each unique row

wspath <- system.file("extdata", package = "wgaim")
genoSxT <- read.cross("csvr", file="genoSxT.csv", genotypes=c("AA","BB"),
  na.strings = c("-", "NA"), dir = wspath)
genoSxT <- cross2int(genoSxT, missgeno="MartinezCurnow", id = "id")

# plot linkage map

link.map(genoSxT, cex = 0.5)

## End(Not run)
```

---

genoCxR

*Genotypic marker data for Cascades x RAC875-2 doubled haploid population in R/qtl format*

---

## Description

Linkage map marker data for the Cascades x RAC875-2 doubled haploid population in the form of an R/qtl object.

## Usage

```
data(genoCxR)
```

## Format

This data relates to a linkage map of 663 markers genotyped on 93 individuals. The linkage map consists of 42 linkage groups spanning the whole genome. Coincident markers have been removed reducing the linkage map to 458 markers. Map distances have been estimated using `read.cross` with the haldane mapping function. The data object is therefore in R/qtl format. See `read.cross` documentation for more details on the format of this object.

## Examples

```
data(genoCxR, package = "wgaim")
link.map(genoCxR, cex = 0.5)
```

---

genoRxK	<i>Genotypic marker data for RAC875 x Kukri doubled haploid population in R/qtl format</i>
---------	--

---

**Description**

Linkage map marker data for the RAC875 x Kukri doubled haploid population in the form of an R/qtl object.

**Usage**

```
data(phenoSxT)
```

**Format**

This data relates to a linkage map of 500 genetic markers genotyped on 368 individuals from the RAC875 x Kukri population. The linkage map consists of 21 linkage groups with varying numbers of markers. Map distances have been estimated using `read.cross` with the `kosambi` mapping function. The data object is therefore in R/qtl format. See `read.cross` documentation for more details on the format of this object.

**Examples**

```
data(genoRxK, package = "wgaim")  
link.map(genoRxK, cex = 0.5)
```

---

genoSxT	<i>Genotypic marker data for Sunco x Tasman doubled haploid population in R/qtl format</i>
---------	--

---

**Description**

Linkage map marker data for the Sunco x Tasman doubled haploid population in the form of an R/qtl object.

**Usage**

```
data(phenoSxT)
```

**Format**

This data relates to a linkage map of 287 genetic markers genotyped on 190 individuals from the Sunco x Tasman population. This set is reduced from the original 345 markers (a mixture of AFLP, RFLP and microsatellite markers and protein analysis). The reduction was created by discarding 58 markers which were co-located with one or more other markers. The linkage map consists of 21 linkage groups with varying numbers of markers. Map distances have been estimated using read.cross with the kosambi mapping function. The data object is therefore in R/qtl format. See read.cross documentation for more details on the format of this object.

**Examples**

```
data(genoSxT, package = "wgaim")
link.map(genoSxT, cex = 0.5)
```

---

<code>link.map.cross</code>	<i>Plot a genetic linkage map</i>
-----------------------------	-----------------------------------

---

**Description**

Neatly plots the genetic linkage map with marker locations and marker names.

**Usage**

```
## S3 method for class 'cross'
link.map(object, chr, chr.dist, marker.names = "markers",
         tick = FALSE, squash = TRUE, m.cex = 0.6, ...)
```

**Arguments**

<code>object</code>	object of class "cross"
<code>chr</code>	character string naming the subset of chromosomes to plot
<code>chr.dist</code>	a list containing named elements "start" and "end" containing the start and end distances in cM the genetic map should be subsetted by. Each of these may also be a vector of distances equal to the length of the number of linkage groups to be plotted.
<code>marker.names</code>	a character string naming the type of marker information to plot. If "dist" then distances names plotted alongside each chromosome on the left. If "markers" then marker names are plotted instead. Defaults to "markers"
<code>tick</code>	logical value. If TRUE then an axis with tick marks are generated for the chromosome names. Defaults to FALSE
<code>squash</code>	logical value. if TRUE then creates extra room on the left side of the chromosomes. This is useful for plotting trait names for QTLs using <code>link.map.wgaim</code> and <code>link.map.default</code>



m.cex	the expansion factor to use for the marker names
...	arguments passed to "plot" to set up the plot region. Arguments may also be passed to "text" for the manipulation of the marker names

### Details

This plotting procedure provides a neater visual display of the chromosomes without marker names overlapping vertically. The plotting region will adjust itself to ensure that all marker names are in the region. For this reason the value for "m.cex" is passed to "text" and should be manipulated until an aesthetic genetic map is reached.

For large maps with many chromosomes, marker names and adjacent chromosomes will overlap horizontally. For the interest of readability this has not been corrected. For this particular situation it is suggested that the user horizontally maximise the plotting window until no overlapping occurs or subset the genetic map to achieve the desired result.

### Value

This invisibly returns the following list for manipulation with `link.map.wgaim`

mt	A list named by the chromosomes with each element containing the locations of the marker names after correcting for overlapping
map	A list named by the chromosomes with each element containing the locations of markers on the chromosomes
chrpos	The numerical position of the chromosomes on the plotting region

### Author(s)

Julian Taylor

### References

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL <http://www.jstatsoft.org/v40/i07/>.

### See Also

[link.map.wgaim](#)

### Examples

```
data(genoSxT, package = "wgaim")

## plot linkage map with marker names

link.map(genoSxT, cex = 0.5)

## plot linkage map with distances
```

```
link.map(genoSxT, cex = 0.5, marker.names = "dist")
```

---

link.map.default      *Plot a genetic linkage map with QTL for multiple traits*

---

### Description

Neatly plots the genetic linkage map with marker locations, marker names and highlights QTL's with their associated flanking markers for multiple traits obtained from a list of `wgaim` fits.

### Usage

```
## Default S3 method:
link.map(object, intervalObj, chr, chr.dist, marker.names
         = "markers", list.col = list(q.col = rainbow(length(object)),
                                     m.col = "red", t.col = rainbow(length(object))), list.cex =
         list(m.cex = 0.6, t.cex = 0.6), trait.labels = NULL, tick = FALSE, ...)
```

### Arguments

<code>object</code>	a list object with elements inheriting the class "wgaim"
<code>intervalObj</code>	object of class "cross" or "interval"
<code>chr</code>	character string naming the subset of chromosomes to plot
<code>chr.dist</code>	a list containing named elements "start" and "end" containing the start and end distances in cM the genetic map should be subsetted by. Each of these may also be a vector of distances equal to the length of the number of linkage groups to be plotted.
<code>marker.names</code>	a character string naming the type of marker information to plot. If "dist" then distances names plotted alongside each chromosome on the left. If "markers" then marker names are plotted instead. Defaults to "markers".
<code>list.col</code>	named list of colors used to highlight the QTL regions and their flanking markers. <code>q.col</code> is the colors of the QTL regions (defaults to <code>rainbow(n)</code> where <code>n</code> is the length of <code>object</code> ). <code>m.col</code> is the color the flanking markers. <code>t.col</code> is the color of the trait names used in each model (defaults to the same color as the QTL regions). See <code>par</code> for color options
<code>list.cex</code>	a named list object containing the character expansion factors for the marker names <code>m.cex</code> and the trait labels <code>t.cex</code>
<code>trait.labels</code>	character string naming the trait used in the model object, defaults to the names of the traits used in each model.
<code>tick</code>	logical value. If TRUE then an axis with tick marks are generated for the chromosome names
<code>...</code>	arguments passed to "plot" to set up the plot region. Arguments may also be passed to "text" for the manipulation of the marker names

## Details

This plotting procedure is a wrapper for `link.map.wgaim` and displays QTL for multiple traits obtained from a list of models given by `object`. Alternative labels for the traits can be given, in model order, using `trait.labels`.

Color specific highlighting of the QTL is also available using `clist`. This differs slightly from `link.map.wgaim`. Here the `q.col` and `t.col` should be given a set of colors equal to the length of `object`. Let `n` be the length of `object`. Then if `q.col` is `NULL` or length of `q.col` is not equal to `n` then it defaults to `rainbow(n)`. If `t.col` is `NULL` or length of `t.col` is not equal to `n` or 1 then it defaults to the colors of `q.col`. Examples of different color combinations are given below.

The `list.cex` argument can be used to manipulate the character expansion of the marker names using `m.cex` or the character expansion of the `trait.labels` using `t.cex`. If a set of "marker" analyses has been performed then `pch` is used to plot a symbol at the location of the QTL. This character can be changed using the usual arguments such as `pch` or `cex` that are passed through the usual `...` argument.

## Value

For a set of "interval" analyses, the genetic linkage map is plotted with shaded QTL regions and highlighted flanking markers. For a set of "marker" analyses, symbols are placed at the QTL locations and the markers are highlighted.

## Author(s)

Julian Taylor

## References

Julian Taylor, Arunas Vebyla (2011). R Package `wgaim`: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL <http://www.jstatsoft.org/v40/i07/>.

## See Also

[link.map.cross](#), [link.map.wgaim](#)

## Examples

```
## Not run:
## fit wgaim models

rktgw.qtl <- wgaim(rktgw.asf, phenoData = phenoRxK, intervalObj = genoRxK,
merge.by = "Genotype", trace = "trace.txt", na.method.X = "include")

rkyld.qtl <- wgaim(rkyld.asf, phenoData = phenoRxK, intervalObj = genoRxK,
merge.by = "Genotype", trace = "trace.txt", na.method.X = "include")

## plot QTL intervals
```

```

# matching rainbow QTL color and trait names, red flanking markers
# (default) and gray background markers.

link.map(list(rktgw.ctl,rkyld.ctl), genoRxK, col = "gray")

# rainbow QTL color and black trait names, red flanking markers
# (default) and gray background markers.

link.map(list(rktgw.ctl,rkyld.ctl), genoRxK, list.col = list(t.col =
"black", m.col = "red"), col = "gray")

# monochromatic plot: gray QTLs, black trait names, black flanking
# markers and gray background markers

link.map(list(rktgw.ctl,rkyld.ctl), genoRxK, list.col = list(q.col =
rep(gray(0.8), 2), t.col = "black", mcol = "black"), col = "gray")

## End(Not run)

```

---

link.map.wgaim

---

*Plot a genetic linkage map with QTL's*


---

### Description

Neatly plots the genetic linkage map with marker locations, marker names and highlights QTL's with their associated flanking markers obtained from a fit to wgaim.

### Usage

```

## S3 method for class 'wgaim'
link.map(object, intervalObj, chr, chr.dist,
         marker.names = "markers", list.col = list(q.col = "light blue",
         m.col = "red", t.col = "light blue"), list.cex = list(t.cex = 0.6,
         m.cex = 0.6), trait.labels = NULL, tick = FALSE, ...)

```

### Arguments

object	object of class "wgaim"
intervalObj	object of class "cross" or "interval"
chr	character string naming the subset of chromosomes to plot
chr.dist	a list containing named elements "start" and "end" containing the start and end distances in cM the genetic map should be subsetted by. Each of these may also be a vector of distances equal to the length of the number of linkage groups to be plotted.
marker.names	a character string naming the type of marker information to plot. If "dist" then distances names plotted alongside each chromosome on the left. If "markers" then marker names are plotted instead. Defaults to "markers".

<code>list.col</code>	named list of colours used to highlight the QTL regions and their flanking markers. <code>q.col</code> is the color of the QTL regions. <code>m.col</code> is the color the flanking markers. <code>t.col</code> is the color of the trait name used in the model object (see par for colour options)
<code>list.cex</code>	a named list object containing the character expansion factors for the marker names <code>m.cex</code> and the trait labels <code>t.cex</code>
<code>trait.labels</code>	character string naming the trait used in the model object
<code>tick</code>	logical value. If TRUE then an axis with tick marks are generated for the chromosome names
<code>...</code>	arguments passed to "plot" to set up the plot region and plot any sybols if required

### Details

This plotting procedure builds on `link.map.cross` by adding the QTL regions to the map and highlighting the appropriate markers obtained from a fit to `wgaim`. If the linkage map is subsetted and QTL regions fall outside the remaining map a warning will be given that the QTL have been omitted from the display.

The `list.col` arguments `q.col`, `m.col` and `t.col` have been added for personal colour highlighting of the QTL regions, flanking markers and trait names. For greater flexibility the procedure may also be given the usual `col` argument that will be passed to the other markers.

The `list.cex` argument can be used to manipulate the character expansion of the marker names using `m.cex` or the character expansion of the `trait.labels` using `t.cex`. If a "marker" analysis has been performed then `pch` is used to plot a symbol at the location of the QTL. This character can be changed using the usual arguments such as `pch` or `cex` that are passed through the `...` argument.

### Value

For an "interval" analysis, the genetic linkage map is plotted with shaded QTL regions and highlighted flanking markers. For a "marker" analysis, a symbol is placed at the QTL locations and the markers are highlighted.

### Author(s)

Julian Taylor

### References

Julian Taylor, Arunas Vebyla (2011). R Package `wgaim`: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL <http://www.jstatsoft.org/v40/i07/>.

### See Also

[link.map.cross](#), [wgaim](#)

## Examples

```
## Not run:
# fit wgain model

yield.qtl <- wgain(yield.fm, phenoData = phenoRxK, intervalObj = genoRxK,
merge.by = "Genotype", trace = "trace.txt", na.method.X = "include")

# plot QTL

link.map(yield.qtl, genoRxK, list.col = list(m.col = "red"), col = "gray")

## End(Not run)
```

---

out.stat	<i>Plot the blups or interval outlier statistics from specified iterations of wgain</i>
----------	---

---

## Description

Plots the interval blups/outlier statistics for specified iterations of wgain. The interval blups/outlier statistics appear as a trace across the genome separated by chromosome and appropriately spaced by their distances.

## Usage

```
out.stat(object, intervalObj, int = TRUE, iter = NULL, chr = NULL, stat
= "os", ...)
```

## Arguments

object	object of class "wgain"
intervalObj	object of class "cross" or "interval"
int	logical value, if TRUE then plot interval outlier statistics. If FALSE then plot chromosome outlier statistics.
iter	numeric value determining which iterations will be plotted
chr	character vector naming the subset of chromosomes to plot. This can only be used when int is TRUE
stat	character string naming the value to be plotted. Default is "os" (outlier statistics). Other option is "blups" for the scaled empirical blups calculated during each iteration
...	arguments passed to "xyplot" or "barchart" (with some restrictions, see Details)

## Details

By default the interval blups/outlier statistics are plotted in separate panels for each iteration in a set layout of 5 rows and one column. This cannot be adjusted and users should not attempt to use the layout argument. Viewing multiple pages can be done by specifying the appropriate iterations using the iter argument.

The set of QTL are obtained from the model and printed on the plot in their appropriate positions in each panel.

## Value

The blups/outlier statistics are plotted in a trellis panel plot.

## Author(s)

Julian Taylor

## References

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL <http://www.jstatsoft.org/v40/i07/>.

## See Also

[tr.wgaim](#), [wgaim](#)

## Examples

```
## Not run:
# fit wgaim model

rkyld.qtl <- wgaim(rkyld.asf, phenoData = phenoRxK, intervalObj = genoRxK,
merge.by = "Genotype", trace = "trace.txt", na.method.X = "include")

# plot QTL interval outlier statistics

out.stat(rkyld.qtl, genoRxK, iter = 1:5, cex = 0.4)

## End(Not run)
```

---

 phenoCxR

*Phenotypic Cascades x RAC875-2 zinc experiment data frame*


---

**Description**

Zinc concentration data of a Doubled Haploid wheat population.

**Usage**

```
data(phenoCxR)
```

**Format**

This data relates to a glasshouse experiment involving a set of 90 Doubled Haploid (DH) lines from a crossing of Cascades x Rac875-2. The DH lines were allocated randomly to pots in the glasshouse using a randomised complete block design. There were also additional pots that contained 5 of each of the parents (Cascades and Rac-875-2). Two measurements were made, namely zinc concentration and shoot length. The data frame consists of 200 rows and 5 columns described below

**id:** A factor of 92 levels containing the unique identification of the DH lines and parents.

**Block:** A factor of two levels indexing the blocks in the experiment

**Type:** A factor of 3 levels indexing the wheat variety (Doubled Haploid, Cascades, Rac875-2)

**shoot:** A numerical variable of shoot lengths for each plant

**znconc:** A numerical variable of zinc concentration levels for each plant

**Examples**

```
data(phenoCxR, package = "wgaim")
plot(phenoCxR$shoot, phenoCxR$znconc)
```

---

 phenoRxK

*Phenotypic RAC875 x Kukri trial data frame*


---

**Description**

Phenotype data arising from a field trial of a Doubled Haploid population involving a crossing of the wheat varieties RAC875 and Kukri

**Usage**

```
data(phenoRxK)
```



**Format**

This data relates to a field trial conducted in 2007 at the Roseworthy Campus of the University of Adelaide. The trial consisted of 2 replicates of 367 Doubled Haploid lines from a cross between wheat varieties RAC875 and Kukri. The DH lines, the parents (RAC875, Kukri) and commercial varieties (Co5693\*E002, DRYSDALE, EXCALIBUR, KRICHAUFF, WYALKATCHEM, YITPI) were randomly allocated to 756 plots using a randomized complete block design. The trial was laid out in a 63 by 12 rectangular array. The data frame consists of 756 rows with 8 columns described by

**Genotype:** A 375 level factor containing a unique identification for the wheat varieties involved in the experiment.

**Type:** A factor of four levels indexing the wheat varieties (Doubled Haploid, RAC875, Kukri, Other) where Other consist of (Co5693\*E002, DRYSDALE, EXCALIBUR, KRICHAUFF, WYALKATCHEM, YITPI)

**id:** A factor of 183 levels uniquely identifying the wheat varieties involved in the experiment.

**Range:** A factor of 12 numeric levels indexing the field Range.

**Row:** A factor of 63 numeric levels indexing the field Rows.

**Block:** A factor of 2 levels indexing the Blocks of the experiment

**yield:** A numeric vector of yield observations in kg/ha

**tgw:** A numeric vector of thousand grain weight observations

**Examples**

```
data(phenoSxT, package = "wgaim")
```

---

phenoSxT

*Phenotypic Sunco x Tasman trial data frame*

---

**Description**

Phenotype data arising from a two-phase experiment involving a Doubled Haploid population from a crossing of the wheat varieties Sunco and Tasman

**Usage**

```
data(phenoSxT)
```

**Format**

This data relates to a two-phase experiment involving a set of 175 Doubled Haploid lines. In the first phase DH lines were randomly allocated to plots using a complete block design with additional plots containing the parents (Sunco, Tasman) as well as commercial lines (Frame, Janz, Krichauff, Machete, RAC820, Trident). The trial was laid out in a rectangular array of 31 rows and 12 columns. In the second phase 23% of the field samples were replicated in the milling process producing a total

of 456 milling samples. These partially replicated field samples were then randomly allocated to 38 mill days with 12 samples per mill day. The data frame consists of 456 rows with 11 columns. These columns are

**Expt:** A one level of factor containing a unique identification for the experiment.

**Type:** A factor of nine levels indexing the wheat variety (Doubled Haploid, Sunco, Tasman, (Frame, Janz, Krichauff, Machete, RAC820, Trident))

**id:** A factor of 183 levels uniquely identifying the wheat varieties involved in the experiment.

**Range:** A factor of 12 numeric levels indexing the field Range.

**Row:** A factor of 31 numeric levels indexing the field Rows.

**Rep:** A factor of 2 levels indexing the Block of the experiment

**Millday:** A factor of 38 numeric levels indexing the milling day

**Millord:** A factor of 12 levels indexing the milling order

**myield:** A numeric vector of milling yield observations from the second phase of the experiment.

**lord:** A centered numerical vector of milling orders, Millord

**lrow:** A centered numerical vector of Rows

### Examples

```
data(phenoSxT, package = "wgaim")
```

---

qt1Table

*Stack QTL summary information into a super table*

---

### Description

Stack QTL summary information into a super table ready for simple exporting

### Usage

```
qt1Table(..., intervalObj = NULL, labels = NULL, columns = "all")
```

### Arguments

ldots	list of objects of class "wgaim". All "wgaim" models must have been analysed with the same genetic type (see <code>wgaim.asreml</code> )
intervalObj	a genetic object of class "interval" usually used in a <code>wgaim</code> fit
labels	a vector of character strings determining the trait names of each QTL table. if this is NULL then the trait names are found from the response of the <code>wgaim</code> model
columns	this can be either a numeric vector determining which columns of the QTL summaries should be outputted or "all" for all columns. The default is "all".
...	a numeric vector determining which columns of the summary should be returned (see Details)

## Details

The super table is created by stacking the QTL summaries on top of each other using the models in ... from left to right. An extra column is created on the left hand side of the stacked table for the trait names given in the `labels` argument. The names are only placed in the first element of each table with NAs for the rest of the elements. This then allows simple exporting to spreadsheet packages or with the R/LaTeX package **xtable**.

## Value

A data.frame object with stacked QTL summaries

## Author(s)

Julian Taylor

## References

Julian Taylor, Arunas Vebyla (2011). R Package `wgaim`: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL <http://www.jstatsoft.org/v40/i07/>.

## See Also

[wgaim](#)

## Examples

```
## Not run:

## fit wgaim models

rktgw.qtl <- wgaim(rktgw.asf, phenoData = phenoRxK, intervalObj = genoRxK,
merge.by = "Genotype", trace = "trace.txt", na.method.X = "include")

rkyld.qtl <- wgaim(rkyld.asf, phenoData = phenoRxK, intervalObj = genoRxK,
merge.by = "Genotype", trace = "trace.txt", na.method.X = "include")

## create super table and export

qtl_t <- qtlTable(rktgw.qtl, rkyld.qtl, labels = c("Conc.", "Shoot"))
print(xtable(qtl_t), file = "superQTL.tex", include.rownames = FALSE)

## End(Not run)
```

summary.wgaim

*Summary and print methods for the class "wgaim"***Description**

Prints a summary of the "wgaim" object in a presentable format

**Usage**

```
## S3 method for class 'wgaim'
summary(object, intervalObj, LOD = TRUE, ...)
## S3 method for class 'wgaim'
print(x, intervalObj, ...)
```

**Arguments**

object	an object of class "wgaim" (see Details)
x	an object of class "wgaim"
intervalObj	a data structure of class "cross" or "interval" containing the genotypic data
LOD	logical value. If TRUE LOD scores for QTL are calculated, defaults to TRUE
...	further arguments passed to or from other methods

**Details**

It is important that the intervalObj is not missing in summary.wgaim or print.wgaim as it contains vital summary information about each of the QTL detected.

The summary of the QTL differs depending on the method chosen in the analysis using wgaim.asreml. If method = "random" then the significance of the QTL are summarized using a probabilistic argument based on the conditional distribution of the QTL sizes given the data (see Verbyla et. al, 2012 in References) Thus, for each QTL, a value is calculated that represents the probability that the QTL size is greater than zero (or less than zero if the effect is negative). If method = "fixed" then the significance of the QTL is summarized using a one degree of freedom Wald statistic.

**Value**

A summary of the QTL component of the "wgaim" object is printed to the screen. For each QTL detected, if an "interval" analysis was performed then summary.wgaim prints which chromosome, name and distance of each flanking marker, size, probability/p-value, contribution of genetic variance and LOD score if desired. If a "marker" analysis was performed then the chromosome, name and distance of the associated marker, size, probability/p-value, contribution of genetic variance and LOD score are printed. print.wgaim provides a narrative brief of the QTL's detected.

**Author(s)**

Julian Taylor, Simon Diffey, Ari Verbyla and Brian Cullis

## References

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL <http://www.jstatsoft.org/v40/i07/>.

Verbyla, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical And Applied Genetics*, **116**, 195-211.

## See Also

[wgaim.asreml](#)

## Examples

```
## Not run:
# read in data

data(phenoRxK, package = "wgaim")
data(genoRxK, package = "wgaim")

# subset linkage map and convert to "interval" object

genoRxK <- subset(genoRxK, chr = c("1A", "2D1", "2D2", "3B"))
genoRxK <- cross2int(genoRxK, missgeno = "Martinez",
                    id = "Genotype", map.function = "kosambi")

# base model

rkyld.asf <- asreml(yld ~ Type + lrow, random = ~ Genotype + Range,
                  rcov = ~ ar1(Range):ar1(Row), data = phenoRxK)

# find QTL

rkyld.qtl <- wgaim(rkyld.asf, phenoData = phenoRxK, intervalObj = genoRxK,
                  merge.by = "Genotype", gen.type = "interval", method = "fixed",
                  selection = "interval", trace = "trace.txt", na.method.X = "include")

# summarise

print(rkyld.qtl, genoRxK)
summary(rkyld.qtl, genoRxK)

## End(Not run)
```

tr.wgaim

*Display diagnostic information about wgaim QTL model***Description**

Displays diagnostic information about QTL detection and significance for the sequence of models used in a wgaim fit.

**Usage**

```
## S3 method for class 'wgaim'
tr(object, iter = 1:length(object$QTL$effects),
    diag.out = TRUE, ...)
```

**Arguments**

object	an object of class "wgaim"
iter	a vector of integers describing the rows of the p-value matrix to display
diag.out	logical value. If TRUE then diagnostic information about the testing of the genetic variance is given for all iterations.
...	arguments passed to <code>print.default</code> for displaying of information

**Details**

By default the printing of the objects occur with arguments `quote = FALSE` and `right = TRUE`. Users should avoid including these arguments.

**Value**

A probability value matrix for the successive QTL selected is displayed with rows according to the iterations specified. If `diag.out = TRUE` then a matrix with rows consisting of the likelihood with genetic variance, the likelihood without genetic variance (NULL model), the statistic and the p-value for the statistic.

**Author(s)**

Julian Taylor

**References**

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL <http://www.jstatsoft.org/v40/i07/>.

**See Also**[wgaim](#)**Examples**

```
## Not run:
# read in data

data(phenoRxK, package = "wgaim")
data(genoRxK, package = "wgaim")

# subset linkage map and convert to "interval" object

genoRxK <- subset(genoRxK, chr = c("1A", "2D1", "2D2", "3B"))
genoRxK <- cross2int(genoRxK, missgeno = "Martinez",
                    id = "Genotype", map.function = "kosambi")

# base model

rkyld.asf <- asreml(yld ~ Type + lrow, random = ~ Genotype + Range,
                  rcov = ~ ar1(Range):ar1(Row), data = phenoRxK)

# find QTL

rkyld.qtl <- wgaim(rkyld.asf, phenoData = phenoRxK, intervalObj = genoRxK,
                  merge.by = "Genotype", gen.type = "interval", method = "fixed",
                  selection = "interval", trace = "trace.txt", na.method.X = "include")

# diagnostic check

tr(rkyld.qtl, digits = 4)

## End(Not run)
```

---

`wgaim.asreml`*wgaim method for class "asreml"*

---

**Description**

Fits an iterative Whole Genome Average Interval Mapping (wgaim) model for QTL detection

**Usage**

```
## S3 method for class 'asreml'
wgaim(baseModel, phenoData, intervalObj, merge.by = NULL,
      gen.type = "interval", method = "fixed", selection = "interval",
      exclusion.window = 20, breakout = -1, TypeI = 0.05, attempts = 5,
      trace = TRUE, verboseLev = 0, ...)
```

**Arguments**

<code>baseModel</code>	a model object of class "asreml" usually representing a base model with which to build the qtl model.
<code>phenoData</code>	a data frame containing the phenotypic elements used to fit <code>baseModel</code> . This data is checked against the base models data
<code>intervalObj</code>	a list object containing the genotypic data, usually an "interval" object obtained from using <code>cross2int</code> . This object may contain many more markers than observations (see Details).
<code>merge.by</code>	a character string or name of the column(s) in <code>phenoData</code> and <code>intervalObj</code> to merge the phenotypic and genotypic data sets.
<code>gen.type</code>	a character string determining the type of genetic data to be used in the analysis. Possibilities are "interval" and "markers". The default is "interval". (see Details).
<code>method</code>	a character string determining the type of algorithm to be used in the analysis. Possibilities are "random" and "fixed". The default is "random". (see Details).
<code>selection</code>	a character string determining the type of selection method that is used to select QTL in the analysis. Possibilities are "interval" and "chromosome". The default is "interval". (see Details).
<code>exclusion.window</code>	For each QTL, the distance in centimorgans around each QTL that is excluded from further analysis.
<code>breakout</code>	A numerical integer equivalent to the iteration the algorithm breaks out. The default is -1 which ensures the algorithm finds all QTL before halting. (see Details)
<code>TypeI</code>	a numerical value determining the level of significance for detecting a QTL. The default is 0.05.
<code>attempts</code>	An integer representing the number of attempts at convergence for the fixed or random QTL model. The default is 5.
<code>trace</code>	An automatic tracing facility. If <code>trace = TRUE</code> then all asreml output is piped to the screen during the analysis. If <code>trace = "file.txt"</code> , then output from all asreml models is piped to "file.txt". Both trace mechanisms will display a message if a QTL is detected.
<code>verboseLev</code>	numerical value, either 0 or 1, determining the level of tracing outputted during execution of the algorithm A 0 value will produce the standard model fitting output from the fitted ASReml models involved in the forward selection. A value of 1 will add a table of chromosome and interval outlier statistics for each iteration.
<code>...</code>	Any other extra arguments to be passed to each of the asreml calls. These may also include <code>asreml.control</code> arguments.

**Details**

In the initial call to `wgaim.asreml`, the marker or interval information is collated from `intervalObj`. If `gen.type = "interval"` then midpoints of intervals are collated from the "intval" component of `intervalObj`. If `gen.type = "markers"` then markers are collated from the



"imputed.data" component of intervalObj (It should be noted that a "marker" analysis is less efficient than an "interval" analysis as it does not take into account the correlation of the marker effects in the specificity of the model; see Verbyla et. al (2007).

The method argument in wgaim.asreml allows the user access to two algorithms. If method = "fixed" the algorithm places selected QTL as an additive set of fixed effects in the model as the forward selection algorithm proceeds. If method = "random" places selected QTL in the random part of the model as an additive set of random effects. This new formulation is outlined in Verbyla et. al (2012).

The selection argument determines the type of selection algorithm for the analysis. If selection = "chromosome" then outlier statistics for each chromosome are calculated and the largest chromosome or linkage group is chosen. The largest marker/interval outlier statistic in this linkage group is then selected as the putative QTL. If selection = "interval", only marker/interval statistics are calculated and the largest marker/interval is chosen as the putative QTL.

Note: If a genetic map has a small number of markers on a linkage group then using selection = "chromosome" as the selection algorithm is known to be flawed (see Verbyla & Taylor, 2012). For this reason it is suggested that this option only be used when there are a moderate number of markers on each linkage group.

Users can now break out of the algorithm using the breakout argument. If a numerical value greater than zero is given, then the forward selection algorithm breaks at the iteration equal to that value and returns the collected information to this point. This includes fixed/random QTL effects, diagnostic components such as interval/marker BLUPs and outlier statistics as well as the trace components of the algorithm. It should be noted that the algorithm breaks out before a QTL has been moved to the fixed/random effects and estimated. Therefore a positive integer, say n will not return an estimate of the nth QTL but it will return whole genome QTL effects and associated diagnostics for the nth iteration.

It is recommended that trace = "file.txt" be used to pipe the sometimes invasive tracing of asreml licensing and fitting numerics for each model to a file. Errors, warnings and messages will still appear on screen during this process. Note some warnings that appear may be passed through from an asreml call and are outputted upon exit. These may be ignored as they are handled during the execution of the function.

This version of **wgaim** allows high dimensional marker information to be analysed. A simple transformation of the collated high dimensional marker set shows that it may be reduced to the number of genetic lines used in the analysis. This transformation is internal to the wgaim.asreml call and users can now expect a considerably large acceleration in the performance of **wgaim**.

## Value

An object of class "wgaim" which also inherits the class "asreml" by default. The object returned is actually an asreml object (see asreml.object) with the addition of components from the QTL detection listed below.

QTL                      A list of components from the significant QTL detected including a character vector of the significant QTL along with a vector of the QTL effect sizes. There are also a number of diagnostic measures that can be found in diag that are used in conjunction with tr.wgaim and out.stat.

**Author(s)**

Julian Taylor, Simon Diffey, Ari Verbyla and Brian Cullis

**References**

Verbyla, A. P & Taylor, J. D, Verbyla, K. L (2012). RWGAIM: An efficient high dimensional random whole genome average (QTL) interval mapping approach. *Genetics Research*. **94**, 291-306.

Julian Taylor, Arunas Vebyla (2011). R Package wgaim: QTL Analysis in Bi-Parental Populations Using Linear Mixed Models. *Journal of Statistical Software*, **40**(7), 1-18. URL <http://www.jstatsoft.org/v40/i07/>.

Verbyla, A. P., Cullis, B. R., Thompson, R (2007) The analysis of QTL by simultaneous use of the full linkage map. *Theoretical And Applied Genetics*, **116**, 95-111.

**See Also**

[print.wgaim, summary.wgaim](#)

**Examples**

```
## Not run:
# read in data

data(phenoRxK, package = "wgaim")
data(genoRxK, package = "wgaim")

# subset linkage map and convert to "interval" object

genoRxK <- subset(genoRxK, chr = c("1A", "2D1", "2D2", "3B"))
genoRxK <- cross2int(genoRxK, missgeno = "Martinez",
                    id = "Genotype", map.function = "kosambi")

# base model

rkyld.asf <- asreml(yld ~ Type + lrow, random = ~ Genotype + Range,
                  rcov = ~ ar1(Range):ar1(Row), data = phenoRxK)

# find QTL

rkyld.qtl <- wgaim(rkyld.asf, phenoData = phenoRxK, intervalObj = genoRxK,
                 merge.by = "Genotype", gen.type = "interval", method = "fixed",
                 selection = "interval", trace = "trace.txt", na.method.X = "include")

## End(Not run)
```

# Index

## \*Topic **datasets**

- genoCxR, 6
- genoRxK, 7
- genoSxT, 7
- phenoCxR, 16
- phenoRxK, 16
- phenoSxT, 17

## \*Topic **hplot**

- link.map.cross, 8
- link.map.default, 10
- link.map.wgaim, 12
- out.stat, 14

## \*Topic **package**

- wgaim-package, 2

## \*Topic **regression**

- cross2int, 4
- qt1Table, 18
- summary.wgaim, 20
- tr.wgaim, 22
- wgaim.asreml, 23

cross2int, 3, 4

genoCxR, 6  
genoRxK, 7  
genoSxT, 7

link.map, 3  
link.map(link.map.cross), 8  
link.map.cross, 8, 11, 13  
link.map.default, 3, 10  
link.map.wgaim, 9, 11, 12

out.stat, 3, 14

phenoCxR, 16  
phenoRxK, 16  
phenoSxT, 17  
print.summary.wgaim(summary.wgaim), 20  
print.wgaim, 26  
print.wgaim(summary.wgaim), 20

qt1Table, 18

read.cross, 5

summary.wgaim, 20, 26

tr(tr.wgaim), 22  
tr.wgaim, 3, 15, 22

wgaim, 3, 13, 15, 19, 23  
wgaim(wgaim.asreml), 23  
wgaim-package, 2  
wgaim.asreml, 21, 23